Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claims 1-25 (cancelled).

Claim 26 (previously presented): A method for processing software instructions comprising:

in an in-order execution machine, decomposing a macroinstruction into a plurality of microinstructions;

forcing the parallel issue of at least two of the plurality of microinstructions simultaneously regardless of conflict checking;

executing all of the plurality of microinstructions simultaneously, in lockstep using functional units in a floating point unit;

determining whether an exception occurs in any of the microinstructions, before writing results of the executing to result registers wherein the determining step is performed prior to any writing step and the method does not write any results to temporary registers;

if an exception occurs in any of the microinstructions, canceling all of the microinstructions and preventing the results of the executing from being written to the result registers; and

if no exception occurs in any of the microinstructions, writing the results of the executing to the result registers.

Claim 27 (previously presented): The method of claim 26, further comprising:

determining whether at least two of the plurality of microinstructions must be issued in parallel.

Claim 28 (previously presented): A computer system comprising:

a processor comprising:

a floating point unit comprising a plurality of functional units adapted to execute microinstructions;

a ROM;

a plurality of floating point registers;

wherein the processor is configured to emulate an instruction set by:

in an in-order execution machine, decomposing a macroinstruction into a plurality of microinstructions;

forcing the parallel issue of at least two of the plurality of microinstructions simultaneously to the functional units regardless of conflict checking;

determining whether an exception occurs in any of the functional units, wherein the determining step is performed prior to any setting step and the method does not set any temporary registers;

setting result registers for results of each of the functional units only if no exception occurs in any of the functional units; and

if an exception occurs in any of the microinstructions, canceling all of the microinstructions and preventing the setting of result registers for all of the functional units.

Claim 29 (previously presented): The computer system of claim 28, wherein the processor is further configured to emulate an instruction set by:

determining whether at least two of the plurality of microinstructions must be issued in parallel.